
HTTPLORD

MA2 Connected Web Server

Usage & Important Notes

Iteration 3, Deliverable
Jan 9th, 2015

Changes: add safari iPhone/iPad kludge setting.

Overview

Integrating MA Lighting Technologies MA2 systems in an club or commercial installation environment can be challenging, so HTTPLord provides a way to control your MA2 installation via HTTP.

HTTP (hyper-text transfer protocol) is most commonly used to serve html web pages from websites like Google to web-browser clients like Chrome or Firefox, on PC's, tablets and mobile phones.

For the professional installers in the architectural lighting sector, many automation controllers like Crestron(C) and AMX(c) products can be programmed to make HTTP requests also.

HTTPLord connects to your MA2 installation via MA2-Link (telnet) and then becomes a web server which can trigger any MA2 command, and also provide feedback, via a web page.

This allows the installation designer or maintainer to create customised web pages (with clients logo and direct buttons for example) that can be used to trigger events on the MA2 or display some customisable feedback to the operator or control system.

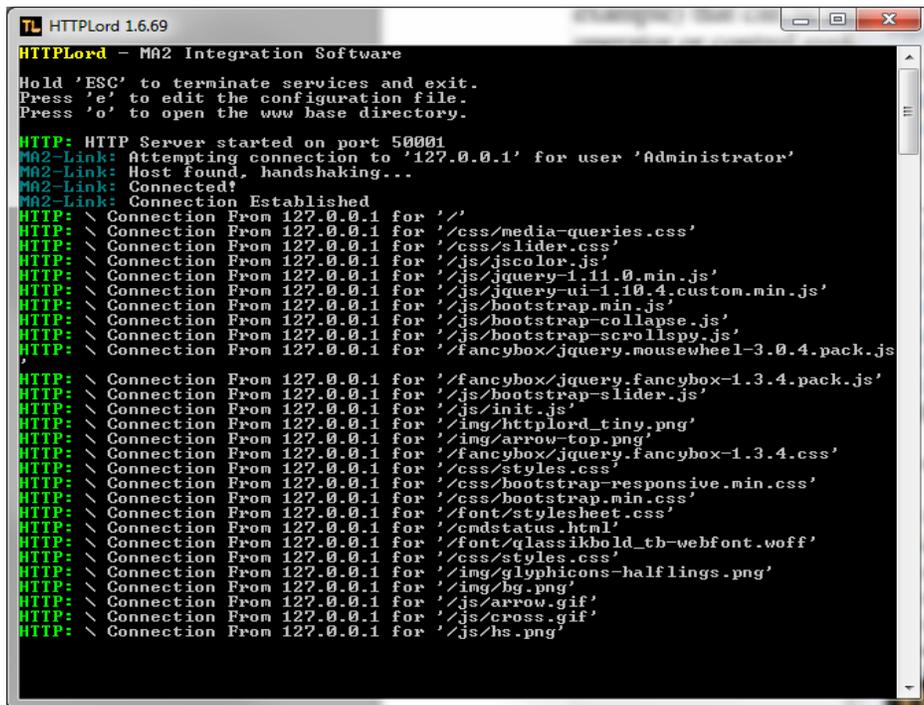
HTTPLord is similar to what web designers would call a CGI, though it is also a basic integrated web server. It can serve up any type of file or html based content also. It is small and fast, and does not have a graphical user interface, instead runs in the background without user intention required.

HTTPLord is © Rowan Maclachlan 2010-2015. All Rights Reserved. All other trademark and brands are property of their respective owners. No pages are intentionally left blank, that's just a waste of paper and bytes.



Features:

- HTTP GET requests
- HTTP POST submits
- Serve and file based content like HTML, images, ccs, javascript etc.
- Example “Mobile Device Remote” web page supplied.
- Real-time Token based HTML substitution.
- Configurable HTML template files for token based generation of Group and Fixture lists
- Automatic reconnection to MA2 if link is lost
- Configuration, html content and templates can be edited whilst running, reducing development time.



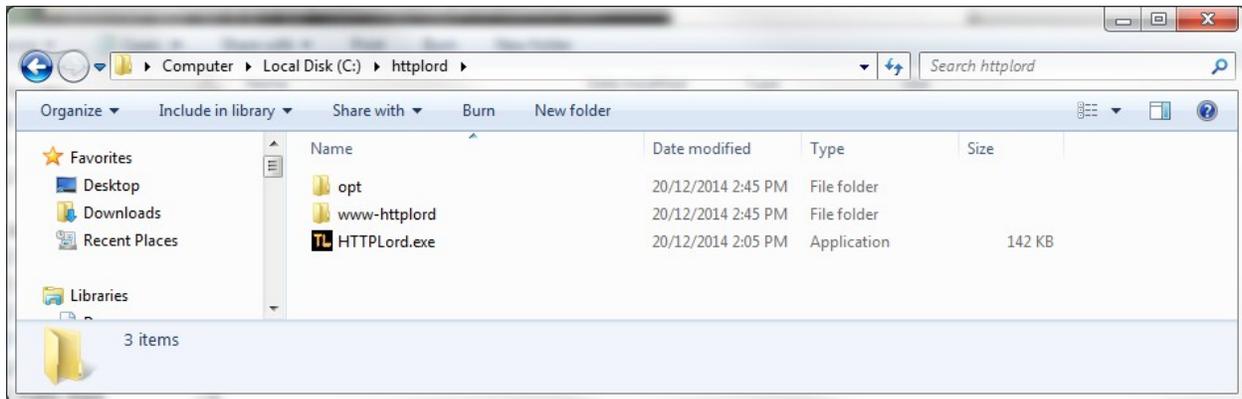
```
HTTPLord 1.6.69
HTTPLord - MA2 Integration Software
Hold 'ESC' to terminate services and exit.
Press 'e' to edit the configuration file.
Press 'o' to open the www base directory.

HTTP: HTTP Server started on port 50001
MA2-Link: Attempting connection to '127.0.0.1' for user 'Administrator'
MA2-Link: Host found, handshaking...
MA2-Link: Connected!
MA2-Link: Connection Established
HTTP: \ Connection From 127.0.0.1 for '/'
HTTP: \ Connection From 127.0.0.1 for '/css/media-queries.css'
HTTP: \ Connection From 127.0.0.1 for '/css/slider.css'
HTTP: \ Connection From 127.0.0.1 for '/js/jscolor.js'
HTTP: \ Connection From 127.0.0.1 for '/js/jquery-1.11.0.min.js'
HTTP: \ Connection From 127.0.0.1 for '/js/jquery-ui-1.10.4.custom.min.js'
HTTP: \ Connection From 127.0.0.1 for '/js/bootstrap.min.js'
HTTP: \ Connection From 127.0.0.1 for '/js/bootstrap-collapse.js'
HTTP: \ Connection From 127.0.0.1 for '/js/bootstrap-scrollspy.js'
HTTP: \ Connection From 127.0.0.1 for '/fancybox/jquery.mousewheel-3.0.4.pack.js'
HTTP: \ Connection From 127.0.0.1 for '/fancybox/jquery.fancybox-1.3.4.pack.js'
HTTP: \ Connection From 127.0.0.1 for '/js/bootstrap-slider.js'
HTTP: \ Connection From 127.0.0.1 for '/js/init.js'
HTTP: \ Connection From 127.0.0.1 for '/img/httplord_tiny.png'
HTTP: \ Connection From 127.0.0.1 for '/img/arrow-top.png'
HTTP: \ Connection From 127.0.0.1 for '/fancybox/jquery.fancybox-1.3.4.css'
HTTP: \ Connection From 127.0.0.1 for '/css/styles.css'
HTTP: \ Connection From 127.0.0.1 for '/css/bootstrap-responsive.min.css'
HTTP: \ Connection From 127.0.0.1 for '/css/bootstrap.min.css'
HTTP: \ Connection From 127.0.0.1 for '/font/stylesheet.css'
HTTP: \ Connection From 127.0.0.1 for '/cmdstatus.html'
HTTP: \ Connection From 127.0.0.1 for '/font/glassikbold_tb-webfont.woff'
HTTP: \ Connection From 127.0.0.1 for '/css/styles.css'
HTTP: \ Connection From 127.0.0.1 for '/img/glyphicons-halflings.png'
HTTP: \ Connection From 127.0.0.1 for '/img/bg.png'
HTTP: \ Connection From 127.0.0.1 for '/js/arrow.gif'
HTTP: \ Connection From 127.0.0.1 for '/js/cross.gif'
HTTP: \ Connection From 127.0.0.1 for '/js/hs.png'
```

HTTPLord example output, showing requests from a web browser.

Installation:

Unzip the contents of the download into a folder you have created and that you have read/write permissions for.



[C:\httplord](#) for example

You can create a shortcut by right-clicking “httplord.exe”. You can then edit the properties of the shortcut to “run minimised” when everything is working and you would like to hide the window automatically. You can add this shortcut to the “Startup” group in the Windows Start menu to run on every time the PC starts.

Configuration:

HTTPLord is configured via an text file called “httplord.ini” file, this is created with defaults the first time you run HTTPLord, for wherever you decide to install it. Once installed, you should not move it around without changing the paths in this file.

You can press 'e' in the console window to open this file for editing. There are several setting which we shall take a look at.

By default, HTTPLord will start a web server on port 50001 (ie: <http://127.0.0.1:50001/>) and try to connect to an MA2 onPC system also on 127.0.0.1, with user “Administrator” password “admin”. It will then begin serving the supplied “Mobile Device Example.”

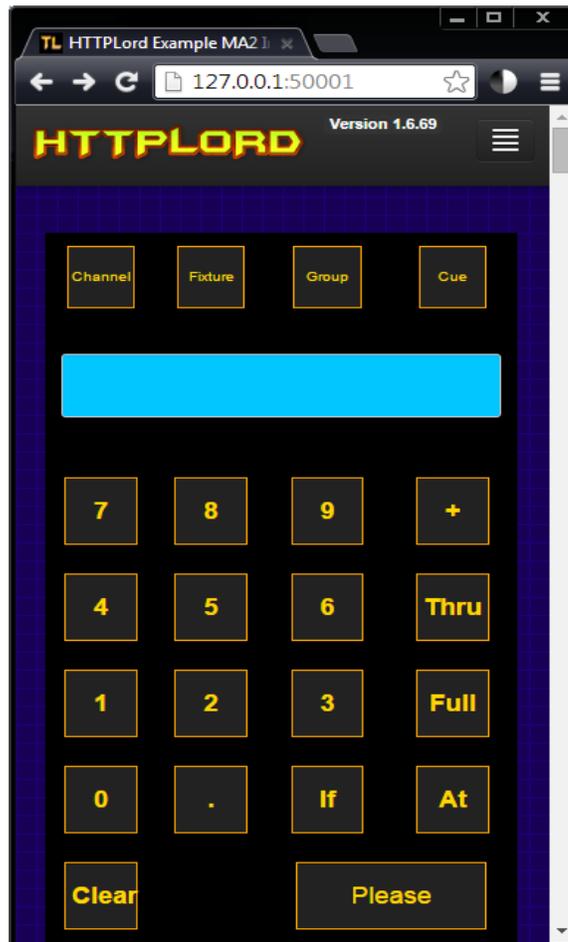
Some Contents and notes on the INI file, comments begin with a ';' character...

```
; --- HTTP Server Settings ---
; Enable the server (1) or disable (0)
www\Enable = 1
; Select the server port (port 80 is for HTTP servers, but others can
be used)
www\Port = 50001
; NOTE: You will need your network firewall configured to allow
incoming TCP access on this port
;
; Base Directory of the server (the path to the directory for storing
html pages and other content)
; by default, this points to the www directory in the folder you
installed HTTPLord.
www\BaseDirectory = C:\HTTPLord\www-httpplord
;
; This is a c:\path\filename.ext that points directly to the mime.txt
file, essential for the server to function.
www\MimeTypeFile = C:\HTTPLord\www-httpplord\mime.txt
; This is a c:\path\filename.ext that points directly to the
Fixture_List_Template.txt file, essential for the {$FIXTURE_LIST}
token function.
www\FixtureListTemplate = C:\HTTPLord\www-
httpplord\fixture_list_template.txt
;
; This is a c:\path\filename.ext that points directly to the
Group_List_Template.txt file, essential for the {$GROUP_LIST} token
function.
www\GroupListTemplate = C:\HTTPLord\www-
httpplord\group_list_template.txt
;
; This is the name of the default page to serve. Typically
'/index.html'
www\IndexHTML = /index.html
;
; This is the name of the default page to serve when a page is not
found. Typically '/404.html'
```

```
www\E404HTML = /404.html
; enable hack to get safari iPhone/iPad "stay on same page" semi
working, no other browser needs this! Very limiting, will reload
index.html after sending a command every time.
www\iSafarikludge = 1
;
; --- MA2 Link Options ---
;
; Enable MA2-Link (must be 1)
; Must enable telnet on console, in Setup \ Console \ Global
ma2\Enable = 1
;
; The IP address of the MA2.
; Try 127.0.0.1 for onPC
ma2\IP = 127.0.0.1
;
; Username and password for an MA2 user, are case sensitive.
ma2\Username = Administrator
ma2\Password = admin
;
; Constantly try to maintain a connection, by auto reconnecting when
connection is lost.
ma2\AutoReconnect = 1
;
; Obtain all MA2 show variables (on first connection) and convert to
Tokens for HTML replacement.
; (read about tokens further on in this document
.ma2\VariablesToTokens = 1
;
; Enable the dynamic token system, variables will be refreshed on
every html page request.
ma2\Variables_EnableDynamic = 1
;
```

So, how do I use it?

To access the example, start “httplord.exe” and open a web browser to “<http://127.0.0.1:50001/>”



You should get a page similar to this. It is designed for portrait mobile devices so resize your browser width. To make it look like this. If you don't get this page, check the HTTPLord window for any error messages and information.

GET method

HTTPLord will respond to HTTP GET commands by returning a “No Content” header, so the browser will stay on the same page.

Example URL: `"/ma2?cmd=Call Macro 1"`

The browser will stay on the same page, unless there is an error. If connected to the MA2, the server will execute the “cmd” parameter, which in this case is “Call Macro 1” (This doesn't work with Safari iPhone/iPad – without the kludge setting mentioned earlier.)

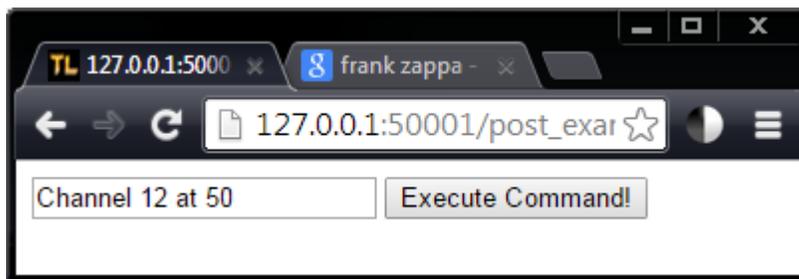
Example HTML: `Fade To Full` will produce a link, labelled “[Fade To Full](#)” which executes the command and also stays on the same page.

POST method

Web designers often use web forms to accept input from users, this example has a text box you can enter text into, and when the form is submitted the “cmdline” value is executed exactly the same way as basic commands described above.

```
<html><head></head><body>
<form id="ma2cmdlinefrm" name="ma2cmdlinefrm" action="/post"
method="post">
<input name="ma2cmdline" id="cmdline" value="" type="text">
  <input type="submit" value="Execute Command!">
</form></body></html>
```

That will produce an input box (press enter to send) and a “Execute Command!” button.



Simple POST example, http://127.0.0.1:50001/post_example.html

Token Replacement

HTTPLord will read through each html file in realtime and replace special keywords with other information, such as lists or variables retrieved from your MA2.

Tokens follow the format of “{ \$ ” at the start, a keyword in all capital letters, and a “} ” at the end. For example, { \$DATE } will be replaced with the date from the MA2 console anywhere it appears in the html file.

HTML replacement tokens built into HTTPLord

{ \$APP_VER }	Current running version of HTTPLord
{ \$MA2_CONNECTED }	Returns “Yes” or “No” depending if an MA2 is connected to HTTPLord
{ \$LAST_RESULT }	Returns “Success” or “Fail” status of the last command sent to the MA2
{ \$LAST_FEEDBACK }	Returns the command response from the last command sent to the MA2. (Only first line)

Dynamic MA2 Variables (defaults for MA2 3.0.0.2)

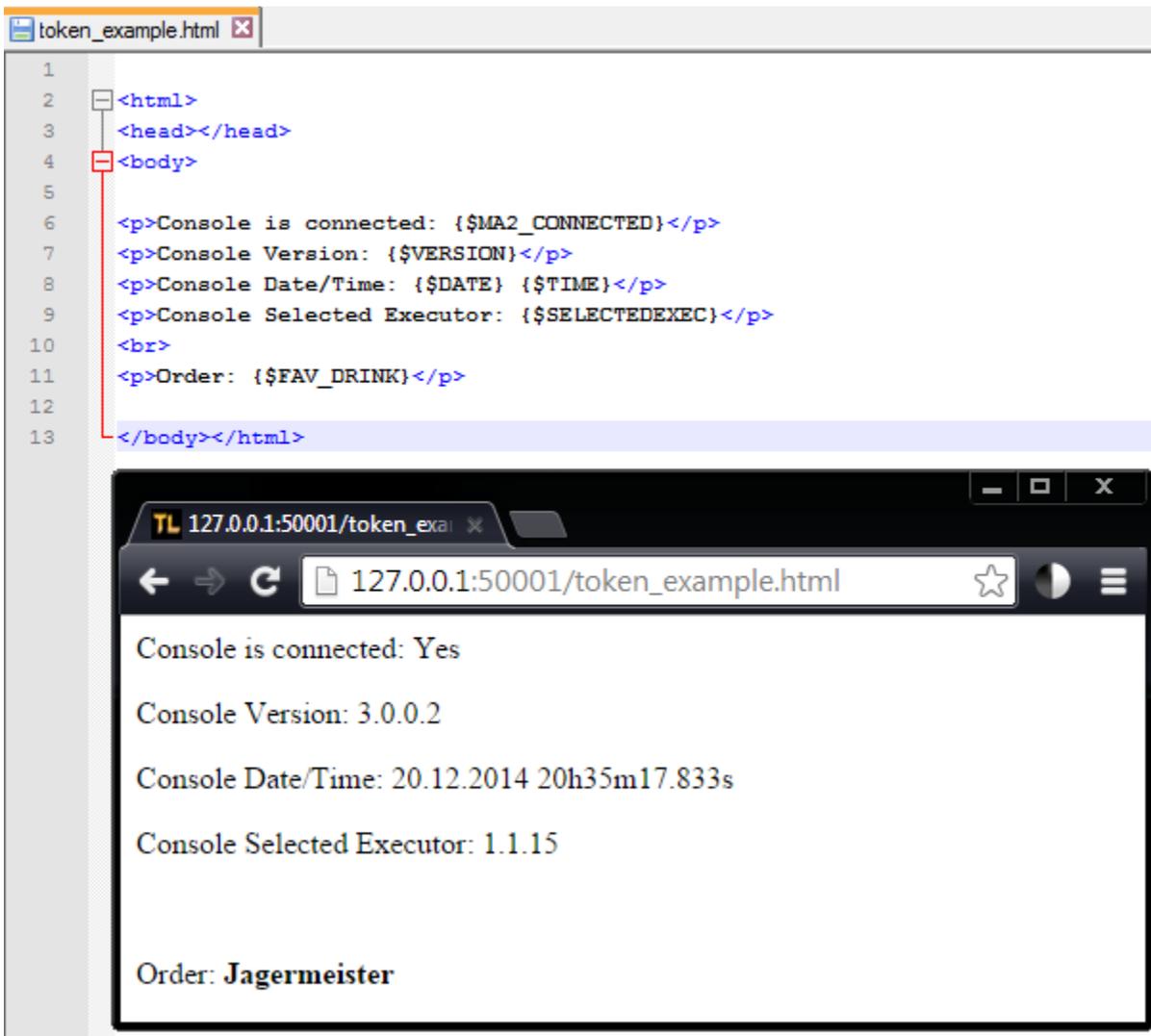
For these to work, the httplord.ini file must contain `ma2\Variables_EnableDynamic = 1`

{ \$DATE }	Current date on the MA2, eg “20.12.2014”
{ \$TIME }	Current time on the MA2, eg “19hr51m46.506s”
{ \$VERSION }	Current connected MA2 software version, eg “3.0.0.2”
{ \$HOSTTYPE }	Current connected MA2 type, eg “Console”
{ \$HOSTSUBTYPE }	Eg, “onPC” “Fullsize” etc.
{ \$PRESET }	Currently selected preset, eg “POSITION”, “DIM”
{ \$SELECTEDEXEC }	Currently selected executor, eg “1.1.15”
{ \$ANY_VARIABLE_NAME }	<p>Returns the value of any MA2 variable, as set by a macro or cue command (or agenda) in the show by the user or console.</p> <p>For example, executing <code>SetUserVar \$INFOMESSAGE="The lights are now on!"</code> the token { \$INFOMESSAGE } in the html would be replaced with the value of the MA2 variable, in this case “The lights are now on” would show on the web page.</p> <p>This is quite powerful feature for installers to modify the values of variables to provide feedback. A scenario might be you have a button on the web page to trigger a lamp on macro. That macro can modify the variable like this <code>SetVar \$ARE_LAMPS_ON=Yes</code> and the lamp off macro can set it to “No”, then the one token { \$ARE_LAMPS_ON } in the html is all that is needed as it's updated everytime the page refreshes.</p> <p>Another scenario is you might have many lighting moods that are selected via a cue list, and the 'cmd' field in the cue list can be used to <code>SetVar \$MOOD=Sexy</code> for the first one, <code>SetVar \$MOOD=Moonlight</code> for the second etc.</p>

Some HTML can also be stored in MA2 variables, for example

```
SetVar $FAV_DRINK="Jagermeister"
```

will make the word appear as bold when displayed, when the token {\$FAV_DRINK} is replaced, as shown in the example below. http://127.0.0.1:50001/token_example.html



Tokens used to feedback information from MA2 variables, the last with embedded HTML.

List Tokens & HTML templates tokens

There are some tokens which are used to generate lists, using template files, to produce a web page with a fixture list for example. This allows for custom formatting of the lists, or use of the html <table> tag for example to create variable sized lists.

Available List Tokens

{ \$GROUP_LIST }	Import the group_list_template file, and produce a copy for each MA2 group that exists. If you have 5 groups for example, the template will be used 5 times.
{ \$FIXTURE_LIST }	Import the fixture_list_template file, and include one for each fixture in the MA2 show.

Group List Token & Template

When a web page includes the token { \$GROUP_LIST }, the template file will be used to display each group that exists in the MA2 show, one by one. This is stored in the group_list_template file, and used to list each group that exists.

Group Template File Tokens

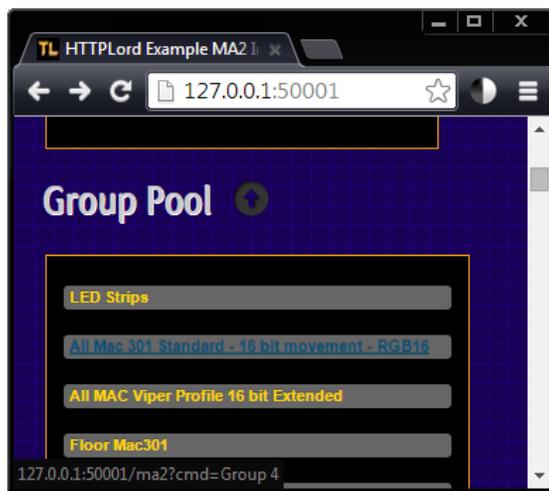
{ \$GROUP_ID }	Replaced with “Group 4” for example
{ \$GROUP_NAME }	Replaced with “Floor Vipers” for example

An example template might be as such:

```
<div><a href="/ma2?cmd={ $GROUP_ID }">{ $GROUP_NAME }</a></div><br>
```

This will, for each MA2 group that exists, make a web link commands the MA2 to select “Group 4” and prints the group name on it like “Floor Mac301”.

Note the
 tag will insert a new line, so all the groups are listed vertically and not on one line.



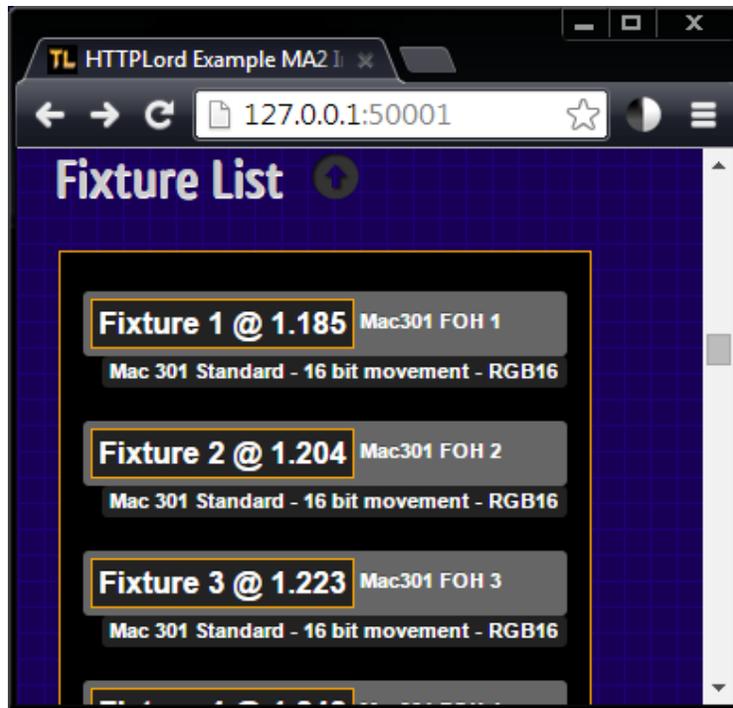
Fixture List Token & Template

When a web page includes the token `{ $FIXTURE_LIST }`, the template file will be used to display each fixture that exists in the MA2 show, one by one. This is stored in the `fixture_list_template` file, and used to list each fixture that exists.

Fixture Template File Tokens

<code>{ \$FIXTURE_ID }</code>	Replaced with “1” for example
<code>{ \$FIXTURE_NAME }</code>	Replaced with “Mac301 FOH 1” for example
<code>{ \$FIXTURE_PATCH }</code>	Replaced with “1.185” for example
<code>{ \$FIXTURE_KEY }</code>	Replaced with “Fixture 1”
<code>{ \$FIXTURE_TYPE }</code>	Replaced with the fixture library type name, eg “Mac 301 Standard – 16 bit movement - RGB16”

Please see the `index.html` and `fixture_list_template.txt` for example code.



Some Important Notes

- HTTPLord requires a TimeLord license for continuous operation.
- We are software programmers and not web designers, and my example is not that great, but you can serve nearly any page/site html template you find on the web using exactly the same principles.
- WE CANNOT SUPPORT YOU FOR HTML/CSS/JAVASCRIPT PROBLEMS – for that you need a web designer. HTTPLord tags and commands should work regardless of the HTML page you are using.
- Malformed or unrecognised tokens will be replaced with {Unknown Token}
- There are javascript examples for the “mobile remote example” in the 'index.html' file.
- Just like an MA2 console or any other critical device, you should probably not allow access to the internet without taking some precautions first.
- Basic security precautions have been taken to prevent access to files from outside the WWW base directory so malicious attackers should not be able to access other files on your computer.
- HTTPLord does not support SSL encryption or .htaccess credentials
- If you require a tightly secured solution, we can provide paid consultation services and custom software, or you can use a proxy solution.
- You could forward this port 50001 on your internet router to the HTTPLord computer to allow remote access from the internet.
- You could implement your own simple password page if you want a password but cannot afford for custom HTTPLord feature development.
- Stopping and starting HTTPLord many times while testing can sometimes lead to MA2 to crash, though this generally only occurs after a dozen stop/start cycles and 'never' happens in typical use.
- HTTPLord by design only accepts one HTTP connection at a time, it is not recommended to have more than 10 users/clients working at one, though depending on the content/settings/utilisation it could be possible to have many more.

Terms and Conditions

Usage of this software ("HTTPLord") is limited to licensed users only (individuals or organisations which have excepted these terms and conditions, and paid the licensing fee to activate functionality). These terms and conditions form a legally binding agreement between you (the user) and TimeLord HQ (Rowan Maclachlan), to which you will adhere at all times. If you do not except these terms and conditions, you must not download TimeLord or purchase an Activation Code, or if you already have you must remove it from your system. The term "Licensing" and "License" are sometimes used to refer to the purchase of an "Activation Code". You, the user or reader of this document agree simply by reading it that any mistakes in these terms and conditions will not void the intentions of the terms and conditions, and understand that these terms and conditions are subject to change. TimeLord HQ has best intentions for you and your use

of the HTTPLord software, but can not offer warranty or guarantee that it will always work as intended, though will attempt to assist you in your efforts through TimeLord HQ's support mechanism. Whilst great effort has been undertaken to build HTTPLord as safe and stable platform, there is always the possibility of a hardware and/or software malfunction or failure, it must be accepted and understood that responsibility for any damages or losses incurred through the use of this software product remain entirely with user. A licensed user may be any individual, organisation, business or corporation, and for the purposes of this document, all users of this software are considered licensed users. Licenses are issued electronically after payment, and licensed on a per-installation basis. Any mentioned brands or products are copyright by their respective owners, and are not authorised for use though used in good faith. Any copyright or patent infringements are completely unintended, and complainants should make immediate contact with the developer (dmxout at gmail dot com), as this software was developed independently for the greater artistic community, and is not the product of a hostile capital venture. Claiming patent infringements on HTTPLord would be a douchebag move, and will be publicised as such and you waive any right to recourse. THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. I DONT REALLY LIKE CAPITAL LETTERS, BUT IT SEEMS TO BE THE LATEST TREND, TIMELORD HQ IS NOT SHOUTING AT YOU. The licensed user agrees to not reverse engineer, disassemble or modify the HTTPLord software components (HTTPLord will not start if it has been modified). The licensed user also agrees to not copy or distribute the HTTPLord software components. License enforcement and payment processing are undertaken by third-party providers, and any problems in service delivery from this supplier are not in the developers control, though every attempt is made to ensure a smooth experience with license enforcement. There is no guarantee that HTTPLord will work with your hardware or software, though every reasonable effort has been made to support a range of 3rd party devices. There is no warranty of fitness for any usage purpose. As the licensed user, you agree that there won't be any legal actions taken against the developer, and acknowledge the premise of the above mentioned statements being toward the resolution of an amicable outcome. HTTPLord can be tested before purchasing an Activation Code, therefore no refunds are available. A license is valid for one installation at a time. During Activation, you must create and enter a password. Failure to produce the correct password on reactivation or deactivation does not a refund make. At most, liability will be limited to the amount tendered for purchase of a individual TimeLord license. HTTPLord developer may at any time choose to cancel your license and subsequently disregard you entirely, especially if your considered to be an ass hole. You agree to not attempt to charge-back your license fee. Otherwise, HTTPLord is copyright in the year of 2012, through to the present day, but has nothing whatsoever to do with the darleks, or the tardis... Modifications to the specification, license, availability, layout and features of HTTPLord should be assumed likely, and at any time can be subject to change without notice. "TimeLord HQ" and "developer" refer to Rowan Maclachlan of Australia. No information is collected and transmitted to TimeLord HQ without your consent, and your privacy is excellent since the only identifiable information about you is stored at PayPal and SoftWorkz (3rd party providers), and is unavailable to me, other than transaction information. You can find the mentioned 3rd party privacy policies online at their respective sites. TimeLord HQ will not assume the responsibility for the loss of anything from anywhere at any time for anybody. I will however accept responsibility for any thing legal that is considered awesome or spectacular in any fashion.

1.7 Contact for Information and Support

For additional information, HTTPLord support can be found through www.timelord-mtc.com or via support@timelord-mtc.com. Please read the manual, release notes and known issues thoroughly before contacting support, as topics already answered in this document or on web-sites may not generate a response from the technical support email address.

WE CANNOT SUPPORT YOU FOR HTML/CSS/JAVASCRIPT PROBLEMS – for that you need a web designer. HTTPLord tags and commands should work regardless of the HTML page you are using.